Intellectual Property

Luminance Histogram

For

Waveform Monitor

Version 1.1

February, 2015

Corporate HQ & Design Center
380 Stevens Ave. Suite 206
Solana Beach, CA 92075
http://www.macnica-na.com

License and Terms of Use

This IP Core with its associated source code and support files, are being provided on an "as-is" basis and as an accommodation. Therefore all warranties, representations or guarantees of any kind (whether express, implied or statutory) including, without limitation, warranties of merchantability, non-infringement, or fitness for a particular purpose, are specifically disclaimed.

This source code may only be used in an Altera programmable logic device and may not be distributed without permission from Macnica Americas, Inc. It is provided free of royalties or fees of any kind.

Document Revision History

Revision	Date	Comments
1.0	August, 2014	Initial Draft
1.1	February, 2015	Minor clean-up and corrections

1 Contents

2 Introduction 4
3 References 4
3.1 Video and Image Processing User Guide, Altera Corp. 4
3.2 Avalon Interface Specifications, Altera Corp. 4
4 Functional Specification 5
4.1 Symbol 5
4.2 Interface Description 5
4.2.1 Control 5
4.2.2 Video 5
4.3 Functional Description 6
4.3.1 Data Sampling 7
4.3.2 Table Organization 8
4.3.3 Overflow 10
4.4 Programming 10
5 Control and Status Registers 11
5.1 Registers 11
6 Specifications 12
6.1 Image Size 12
6.2 Performance 12
7 Resource Utilization 12

2 Introduction

The Luminance Histogram collects samples of an Avalon-ST video stream consisting of a single component. Data is collected in a columnar fashion so that a histogram for several columns of the video is available. This data is usually plotted as a Waveform Monitor that shows the lighting profile of the frame.

3 References

3.1 Video and Image Processing User Guide, Altera Corp.

www.altera.com/literature/ug/ug_vip.pdf

3.2 Avalon Interface Specifications, Altera Corp.

www.altera.com/literature/manual/mnl_avalon_spec.pdf

4 Functional Specification

4.1 Symbol

The QSYS component is shown below. There is one parameter called “SMALL_MEMORY” that can be either true (1) or false (0). The histogram table organization is discussed later in this document. When the parameter SMALL MEMORY is true, there are only 128 bins for each column in the image; when the parameter is false there are 256 bins for each column histogram.

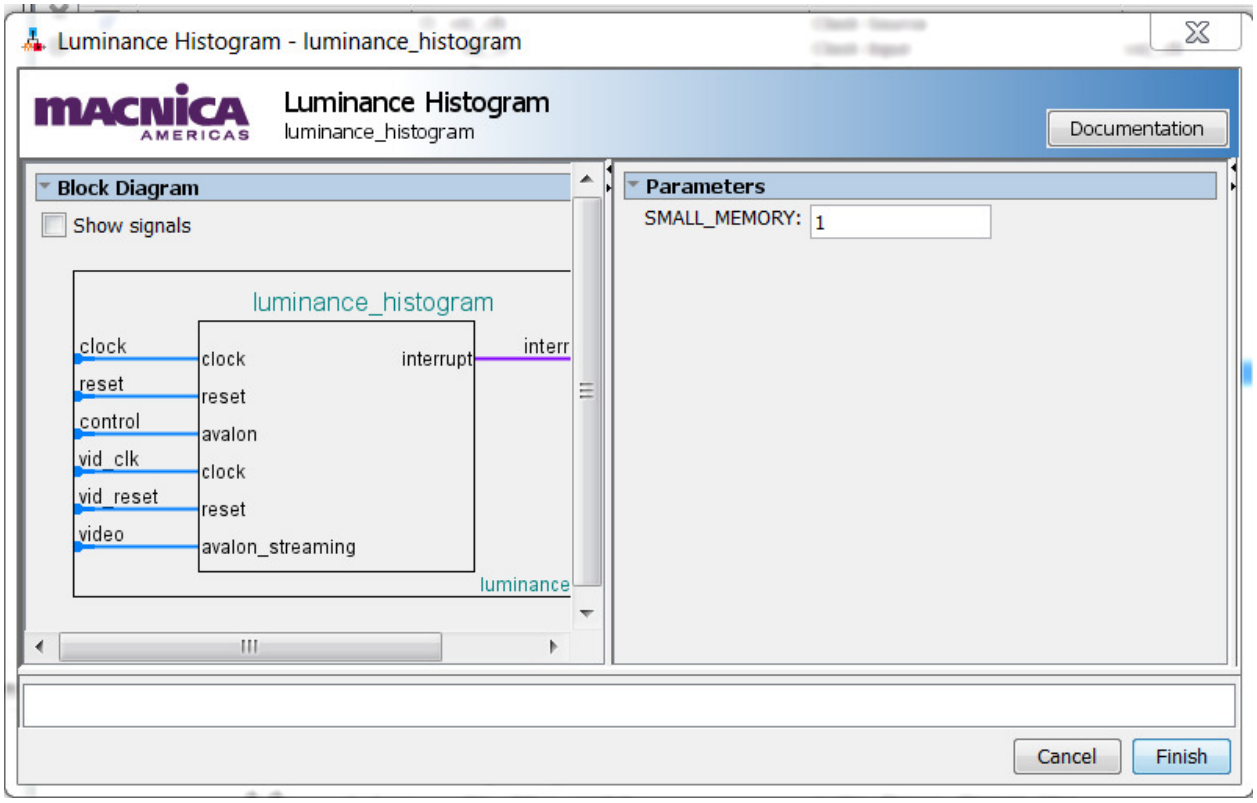


Figure 1 QSYS Component

4.2 Interface Description

4.2.1 Control

Interface	Description	Notes
Avalon Slave	A standard Avalon Slave interface for controlling the various settings and reading the histogram table.	

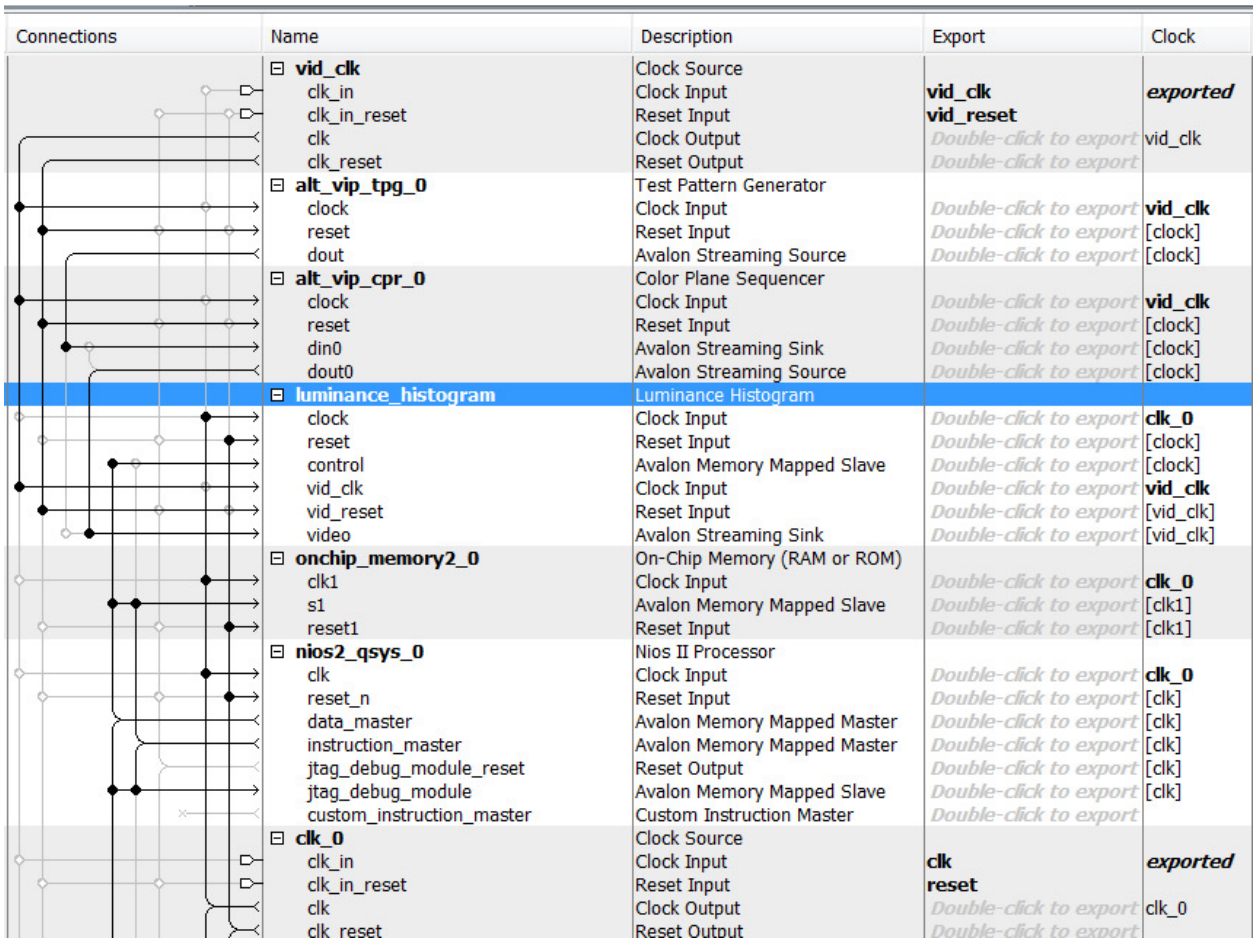
4.2.2 Video

Interface	Description	Notes
Avalon Streaming (sink)	A standard Avalon-ST Video interface, 8-10 bits of luminance data	8bpp – 10bpp parameterized

4.3 Functional Description

The Component Histogram is not a pass-through implementation and has no Avalon Streaming video Source output. Instead, it relies on a Color Plane Sequencer to split off the Y luminance component.

A sample verification testbench implementation is shown below.



Connections	Name	Description	Export	Clock
	vid_clk	Clock Source		
	clk_in	Clock Input	**vid_clk**	***exported***
	clk_in_reset	Reset Input	**vid_reset**	
	clk	Clock Output	*Double-click to export*	vid_clk
	clk_reset	Reset Output	*Double-click to export*	
	alt_vip_tpg_0	Test Pattern Generator		
	clock	Clock Input	*Double-click to export*	**vid_clk**
	reset	Reset Input	*Double-click to export*	[clock]
	dout	Avalon Streaming Source	*Double-click to export*	[clock]
	alt_vip_cpr_0	Color Plane Sequencer		
	clock	Clock Input	*Double-click to export*	**vid_clk**
	reset	Reset Input	*Double-click to export*	[clock]
	din0	Avalon Streaming Sink	*Double-click to export*	[clock]
	dout0	Avalon Streaming Source	*Double-click to export*	[clock]
	luminance_histogram	Luminance Histogram		
	clock	Clock Input	*Double-click to export*	**clk_0**
	reset	Reset Input	*Double-click to export*	[clock]
	control	Avalon Memory Mapped Slave	*Double-click to export*	[clock]
	vid_clk	Clock Input	*Double-click to export*	**vid_clk**
	vid_reset	Reset Input	*Double-click to export*	[vid_clk]
	video	Avalon Streaming Sink	*Double-click to export*	[vid_clk]
	onchip_memory2_0	On-Chip Memory (RAM or ROM)		
	clk1	Clock Input	*Double-click to export*	**clk_0**
	s1	Avalon Memory Mapped Slave	*Double-click to export*	[clk1]
	reset1	Reset Input	*Double-click to export*	[clk1]
	nios2_qsys_0	Nios II Processor		
	clk	Clock Input	*Double-click to export*	**clk_0**
	reset_n	Reset Input	*Double-click to export*	[clk]
	data_master	Avalon Memory Mapped Master	*Double-click to export*	[clk]
	instruction_master	Avalon Memory Mapped Master	*Double-click to export*	[clk]
	jtag_debug_module_reset	Reset Output	*Double-click to export*	[clk]
	jtag_debug_module	Avalon Memory Mapped Slave	*Double-click to export*	[clk]
	custom_instruction_master	Custom Instruction Master	*Double-click to export*	
	clk_0	Clock Source		
	clk_in	Clock Input	**clk**	***exported***
	clk_in_reset	Reset Input	**reset**	
	clk	Clock Output	*Double-click to export*	clk_0
	clk_reset	Reset Output	*Double-click to export*	

Figure 2 - QSYS test system

4.3.1 Data Sampling

Data is collected in a columnar fashion, and the width of the columns is programmable by writing to register 0x03. Valid column widths supported are 4, 8 or 16. All of the pixels in the column width are summed, and the average value is used as the index to the histogram table for the respective column. When the SMALL_MEMORY option is selected, the upper 7 bits of the 8-bit table index to the respective intensity bin are used; when the SMALL_MEMORY option is not selected, there are 256 bins in the table and 8 bits of the averaged intensity value are used. See the figures below.

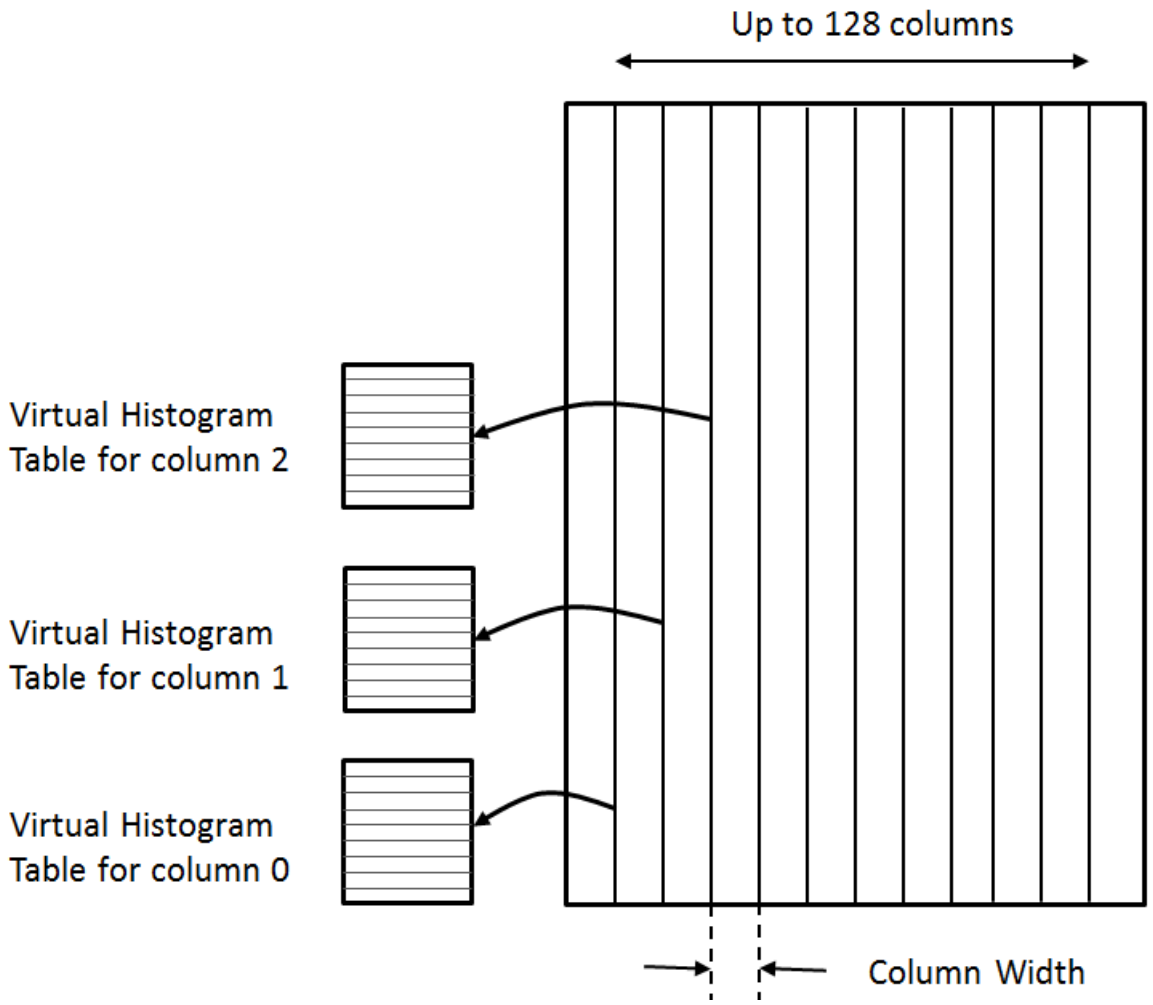


Figure 3 Data Sampling

4.3.2 Table Organization

The histogram table is implemented as a single memory component. The most significant address bits select the column and the least significant bits select the histogram "bins" which contain the count for that bin's intensity value.

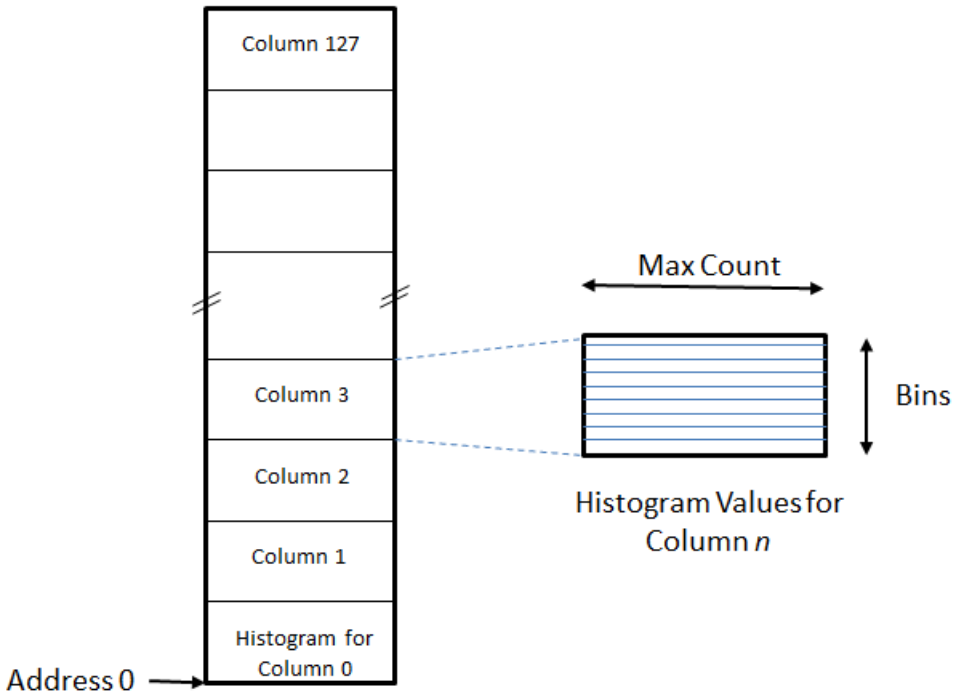


Figure 4 Histogram Table Architecture

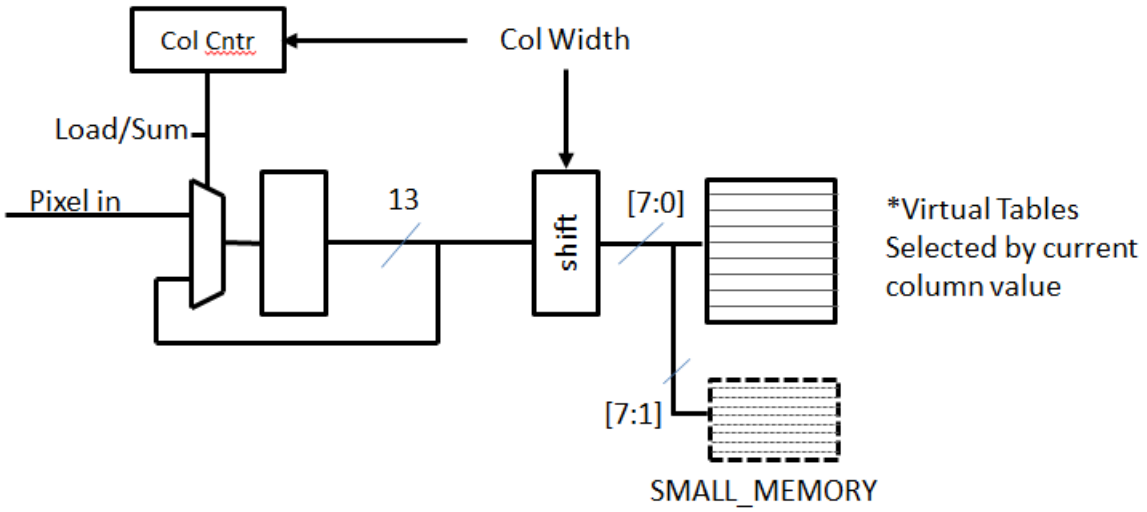


Figure 5 Virtual Table Addressing

Table 1 Histogram Table Implementation Options

Version	Columns	Max Count / Tbl Width	# Bins	Total Memory	M10K
Small Memory	128	1023 / 10 bits	128	16K x 10 bits	16
Large Memory	128	1023 / 10 bits	256	32K x 10 bits	32

4.3.3 Overflow

Since there are 10 bits used for the count in each bin, the maximum count value is 1023. Thus with an image composed of more than 1023 lines it would be possible to overflow the bin, however logic will detect the maximum value in a bin when updating and will prevent overflow from occurring.

4.4 Programming

The programming flow is fairly simple:

1) Enable the Histogram by setting the "go" bit, and optionally the Interrupt Enable bit, in the Control register. The next single valid frame that is received will be analyzed.
2) The host then either waits for the completion interrupt, or checks the "frame complete" bit in the status register.
3) The host clears the interrupt bit (if used)
4) The host reads the contents of the table.
5) The host initiates the clearing of the table by writing to register (2) with any value.
6) The host writes a '0' to bit zero of the status register to enable another capture, or alternatively turns off the "go" bit to disable the function
7) The clearing of the histogram ram requires two clocks per location, so with this 16K-deep table approximately 33,000 clocks are required. This amounts to about 16 lines of incoming HD video, so it is very feasible to analyze every-other video frame. The frame complete and "go" bits are not evaluated until after the histogram table is cleared, so there is obviously ample time for the host CPU to set these bits as desired.

A very simple polled implementation is shown below:

```
#include "sys/alt_stdio.h"
#include <io.h>
#include "../luma_test_bsp/system.h"

int main()
{
  alt_putstr("Hello from Nios II!\n");
int frame_done,i,temp;
  /* Event loop never exits. */
  while (1){
            // set the column width
            IOWR(LUMINANCE_HISTOGRAM_BASE,3,16);  // set column width to 7

            // enable histogram
            IOWR(LUMINANCE_HISTOGRAM_BASE,0,1);
            frame_done = 0;
            while(!frame_done){
                     frame_done=IORD(LUMINANCE_HISTOGRAM_BASE,1);
            }
            //read the table
            for(i=0;i<32;i++)
                     temp = IORD(LUMINANCE_HISTOGRAM_BASE,2);

            //alt_printf("data %x\n",temp);
            // clear it
            IOWR(LUMINANCE_HISTOGRAM_BASE,2,1); // write initiates table clear

            //NOW clear the frame completed bit.  The init_busy will prevent a new capture
            IOWR(LUMINANCE_HISTOGRAM_BASE,1,0);
  }

  return 0;
}
```

5 Control and Status Registers

5.1 Registers

Register	Description	Notes
Control Register (address 0)	This is the standard VIP control register. As usual, bit '0' is the "Go" bit that enables the module. Additional bits are: [0] – The "go" enable bit [1] – interrupt enable	
Status (address 1)	[0] = '1' = frame completed. Writing a '0' to this bit will enable the capture of another frame if the "go" bit is still '1'. [1] = interrupt pending. Writing a '0' to this bit will clear the interrupt.	
Histogram Data (address 2)	Hist Clear [0] = R/W. Clear the histogram table contents by writing a '1' to this bit Writing to this register with any value also clears the auto-incrementing host address to the table.	
Column Width (address 3)	Column Width[4:0] – R/W. This is the column width. Values of 4,8,16 are supported.	

6 Specifications

6.1 Image Size

The implementation supports an image width of 2K pixels -- there are 128 virtual tables that can be based on a column width of 16 pixels.

Since the width of the histogram table is 10 bits, the maximum value for a given intensity “bin” is 1023. Thus, an image height maximum of 1024 is implied since images of greater height could saturate bins and cause undesired clipping in the histogram.

Overflow is prevented by virtue of saturation in each bin. Once the maximum value of 1023 is reached in a given bin it will remain and will not roll-over to 0 if an additional increment to the bin is attempted.

6.2 Performance

The design supports a pixel rate clock of > 188 MHz and a control port clock of > 150 MHz in the test system shown, with a NIOS/e class selected as the control master.

There are no unique timing constraints required.

Slow 1100mV 85C Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	67.76 MHz	67.76 MHz	altera_reserved_tck	
2	150.08 MHz	150.08 MHz	avs_clk	
3	188.61 MHz	188.61 MHz	vid_clk	

7 Resource Utilization

Target Device Family	Variation	Memory	ALM
Cyclone V	SMALL_MEMORY true	163840 bits / 16 M10K	~175
Cyclone V	SMALL_MEMORY false	327680 bits/ 32 M10K	~300